

5

PATENT APPLICATION

METHOD FOR RESIDUAL FORM IN MOLECULAR MODELING

10

Inventor:

Dan E. Rosenthal, a citizen of the United States of America, residing at:
718 Edge Lane
Los Altos, California 94024

15

Assignee:

Protein Mechanics, Inc.
278 Hope Street, Suite C
Mountain View, California 94041

20

Entity:

Small

METHOD FOR RESIDUAL FORM IN MOLECULAR MODELING

CROSS-REFERENCES TO RELATED APPLICATIONS

This application is entitled to the benefit of the priority filing date of
5 Provisional Patent Application No. 60/245,731, filed 2000 Nov. 2, and in addition, co-
pending Provisional Patent Application No. 60/245,730, filed 2000 Nov. 2; No. 60/245,688,
filed 2000 Nov. 2; and No. 60/245,734, filed 2000 Nov. 2; all of which are hereby
incorporated by reference in their entirety for all purposes.

10 BACKGROUND OF THE INVENTION

The present invention is related to the field of molecular modeling and, more
particularly, to computer-implemented methods for the modeling large molecules in which
accelerations appear in the formulation.

The motion of bodies are determined by Newton's Laws of Motion. For a
15 body subject to a force, Newton's Second Law:

$$F = ma$$

or the acceleration of the body of mass is equal to the total force upon the body is applicable.
This simple equation hides enormous complexity for the dynamic modeling and static
analysis of large molecules. The acceleration of the body is the time derivative of velocity of
20 the body and to determine the velocity of the body, its acceleration must be integrated with
respect to time. Likewise, the velocity of a body is the time derivative of position of the body
and to determine the position of the body, its velocity must be integrated with respect to time.
Thus with knowledge of the force upon a body, integration operations must be performed to
determine the velocity and position of the body at a given time.

25 In a molecule, there are multiple bodies whose motions must be considered.
Each body, an atom or collection of atoms, of the molecule is subject to multiple and
complex forces. Thus the calculation of the motion and the shape of the molecule requires
the determination of the position and motion of each atom of the molecule. Hence the
calculation of the structure, dynamics and thermodynamics of molecules, including complex
30 molecules having thousands of atoms, by computers would seem to be the perfect answer.

Indeed, the field of molecular modeling has successfully simulated the motion
(molecular dynamics or MD) and the rest states (static analysis) of many complex molecular
systems by computers. Typical molecular modeling applications have included enzyme-

ligand docking, molecular diffusion, reaction pathways, phase transitions, and protein folding studies. Researchers in the biological sciences and the pharmaceutical, polymer, and chemical industries are beginning to use these techniques to understand the nature of chemical processes in complex molecules and to design new drugs and materials accordingly.

5 Naturally, the acceptance of these tools is based on several factors, including the accuracy of the results in representing reality, the size of the molecular system that can be modeled, and the speed by which the solutions are obtained. The accuracy of the solutions are generally accepted. However, the use of these tools up to now has required enormous computing power to model molecules or molecular systems of even modest size or to obtain molecular
10 time histories of sufficient length to be useful.

There are two sources of computational complexity for most molecular modeling simulations of a large molecule:

1. The particular molecular model which is used to describe the locations, velocities and mass properties of the constituent atoms, the inter-atomic forces between
15 them, and the interactions between the atoms and their surrounding environment: and.
2. The particular numerical method used to advance the model through time. Time is advanced repeatedly by very short intervals, called timesteps, until a final time has been reached.

20 Substantial work has been completed in reducing the computational load for molecular models, such as the reduction of model complexity by constraining higher order modes with rigid body assumptions, Order(N) dynamics, and multi-pole methods for the force field models (see, for example, U.S. Patent No. 5,424,963 on the commercial MBO(N)D software package). The typical formulation method in which the molecular
25 model computes the accelerations of constituent masses is called the "Direct Form" of the equations of motion.

The present invention is directed toward the improvements in the molecular model. This invention provides for an alternative formulation of the molecular model so that fewer computations are required to reach the same result. This alternative method is known
30 as the "Residual Form" of the equations of motion. In this method, only an error, or residual, is calculated such that driving this error to zero ensures that the equations of motion are satisfied. Related methods have been described for use with some numerical integration methods, and in conjunction with mechanical system simulations. For example, see Von Schwerin, *Multibody System Simulation*, Springer, 1999. One example of prior art that uses

the Residual Form include commercial mechanical engineering code sold as SD/FAST. This software provides a Residual Form of the equations (M. Hollars, *et. al.*, *SD/FAST User's Manual*, Version B.2, 1994, p. R-15). Another example includes DAE methods using implicit integration, for example, the DASSL code (Brenan, *et. al.*, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, North-Holland, New York, 1989, Chapter 5), which expects a Residual Form of the equations to be integrated. Also, residual formulation for mechanical multibody simulations is discussed in A. Eichberger, *et. al.*, "The Benefits of Parallel Multibody Simulations" in *International Journal for Numerical Methods in Engineering*, Vol. 37, pp. 1557-1572, 1994.

Attempts have been made to apply residual (or error) functions rather than direct computation of state derivatives to the integration problem when using implicit integration methods. These attempts did not lead to any practical success, in large part because the mechanical systems to which the method were applied were highly cyclic in nature (cyclic meaning many closed loops in the system topology). This necessitated the introduction of additional algebraic quantities into the system description and this complication led to poor conditioning of the equation which caused failure of a certain numerical step central to implicit integrators.

However, molecular models are almost entirely acyclic (very few or no loops in the system topology), and the additional algebraic variables do not need to be introduced into the system model. In the present invention the Residual Form is able to provide a significant speedup to a portion of the computation with a much simpler formulation of the molecular model and its equations of motion. It is believed that Residual Form has never been used in conjunction with molecular modeling and in particular, MD simulations, primarily because MD simulations are usually devised to use explicit numerical methods to advance the molecular model through time, whereas the Residual Form requires the use of implicit numerical methods (see co-pending U.S. Application No. _____, entitled "METHOD FOR LARGE TIMESTEPS IN MOLECULAR MODELING" and filed on even date, which claims priority from the previously referenced provisional patent applications, and which co-pending application is incorporated by reference in its entirety).

The present invention teaches the application of the Residual Form to molecular modeling. By casting MD equations in Residual Form rather than the standard Direct Form, the number of computer operations required to calculate the non-force terms of these equations are drastically reduced. The computational cost of function evaluations in an MD simulation is a significant fraction of the whole simulation and anything that might

reduce this cost is potentially of great benefit. For the preferred embodiment using an Order(N) torsion-angle model with simple active forces, the operation count is reduced by approximately a factor of seven. Of course, the effect on the entire simulation time depends upon the relative costs of the other portions of the computation, and will always be less

5 dramatic than the effect on the model cost alone.

SUMMARY OF THE INVENTION

The present invention teaches a method of computer modeling the behavior of a molecule. The method comprises selecting a model for the molecules, the model having
10 equations of motion for the molecules; formulating the equations of motion in Residual Form; and integrating the model equations with an implicit integrator, to reduce the computer calculations for the molecular behavior. The equations of motion in Residual Form comprise

$$\begin{pmatrix} \rho_q \\ \rho_u \end{pmatrix} = \begin{pmatrix} \dot{q} - W(q)u \\ M(q)\dot{u} - f(t, q, u) \end{pmatrix}$$

where q represents generalized system coordinates, u represents generalized velocities, W
15 represents a generalized joint map matrix, M represents generalized system mass, and f represents generalized system forces. The selected model preferably comprises an Order(N) torsion-angle, rigid multibody system, such as a model with a plurality of rigid bodies, each rigid body representing a portion of the molecule; and a plurality of hinge connections, each hinge connection defining the allowable relative motion between two of the rigid bodies.

20 The present invention also provides for computer code for modeling the behavior of a molecule. The code comprises a model module for the molecules with equations of motion in Residual Form for the molecules; and module having an implicit integrator for integrating the model equations over time to reduce the computer calculations to model the molecular behavior. The model module is preferably for an Order(N) torsion-angle, rigid
25 multibody system.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a representational block module diagram of the software system architecture in accordance with the present invention;

30 Fig. 2 illustrates the tree structure of the multibody system of the molecular model according to the present invention;

Fig. 3 illustrates the reference configuration of the Fig. 2 multibody system;

Fig. 4A illustrate a sliding joint between two bodies of the Fig. 2 multibody system; Fig. 4B illustrate a pin joint between two bodies of the Fig. 2 multibody system; Fig. 4C illustrate a ball joint between two bodies of the Fig. 2 multibody system;

Fig. 5 summarizes general computational steps for the Residual Form method and Direct Form methods of the molecular dynamics computations; and

Fig. 6 is a table which compares the approximate number of computations required for the Direct Form vs. the Residual Form methods for several exemplary MD models.

DESCRIPTION OF THE SPECIFIC EMBODIMENTS

To solve ordinary differential equations (ODEs), most of the prior art have used equations expressed in the Direct Form, i.e., $\dot{y} = f(y, t)$ (where \dot{y} means $\frac{dy}{dt}$). The equations of motion for a biomolecular system can be cast into this form (and called the Direct Form). In molecular modeling, all prior art known to the present inventors have used the Direct Form. That is, $\dot{q} = Wu$, $\dot{u} = M^{-1}f$, where q and u are generalized coordinates and speeds respectively, so that conventional ODE solution methods can be applied. However, this requires a matrix inversion of M (representing the mass of the system) at a cost of $\text{Order}(N)$ to $\text{Order}(N^3)$ floating point operations (depending on algorithm used, where N is the number of degrees of freedom in the system), since the natural form of the equations gives rise to inertial coupling between the derivatives of the generalized speeds. That is, the equations are most naturally produced in the form $\dot{q} - Wu = 0$, $M\dot{u} - f = 0$, where M , the mass matrix, depends explicitly upon the generalized coordinates q , i.e., $M = M(q)$. This fact requires forming and effectively factoring the mass matrix each time the state derivatives are needed by the integrator in integrating the equations of motion over time. The generalized joint map matrix W is block diagonal and, although it is also dependent on the coordinates $W = W(q)$, it does not have a significant computational cost.

In accordance with the present invention, a method for the solution of the equations of a molecular system is expressed in Residual Form to bypass the customary step of producing the state derivatives directly. The Residual Form method has the following steps:

- 1) Discretization of the solution variables. The specific form of discretization is dictated by the particular implicit integration method used to advance the molecular model in

time. Implicit integration follows from the Residual Form. Implicit integration, especially L-stable integrators and other highly stable integrators, such as implicit Euler, Radau5, SDIRK3, SDIRK4, other implicit Runge-Kutta methods, and DASSL or other implicit multistep methods, also provide other advantages for molecular modeling. See, for example, the above-cited U.S. Patent Appln.No. _____, entitled "METHOD FOR LARGE TIMESTEPS IN MOLECULAR MODELING," filed of even date. As a particularly simple example, when used with implicit Euler integration, the discretization is as follows:

$$\dot{q} = (q_n - q_{n-1})/h, \quad \dot{u} = (u_n - u_{n-1})/h$$

where h is the timestep.

2) Substitution into the residual equations:

$$\begin{pmatrix} \rho_q \\ \rho_u \end{pmatrix} = \begin{pmatrix} \dot{q} - W(q)u \\ M(q)\dot{u} - f(t, q, u) \end{pmatrix}$$

3) Solution of the resulting nonlinear algebraic equations $\begin{pmatrix} \rho_q \\ \rho_u \end{pmatrix} = 0$ for q_n and u_n .

The kinematic residual ρ_q compares an estimated \dot{q} generated from the implicit integrator to the derivatives computed by the routines for determining the joints of the molecular model, which is described in greater detail below. The second row of the residual is ρ_u , the dynamic residual, which determines the degree to which an estimated \dot{u} satisfies the equations of motion.

The system mass matrix M and the so-called 'bias-free hinge torque' f are both state dependent. The bias-free hinge torque is generated by the dynamic residual routine when the calculated \dot{u} vector passed to the residual routine is zero. In general, the hinge accelerations are a response to applied forces, joint torques, and motion-induced effects (such as Coriolis and centrifugal forces.) If the system were at rest, and subjected only to joint torques, it would be considered in a bias-free state. The real system with its actual inputs can be reduced to a bias-free state by computing a set of joint torques equivalent to the biased inputs. Both sets produce the same hinge accelerations.

The preferred embodiment of the Residual Form is shown for an Order(N) torsion-angle, rigid-body for the molecular model. The following sections develop the molecular model from basic definitions and show how the model is used to compute the motion of the model. First, the overall computer code architecture for the molecular model

simulation is described. Then an Order(N) torsion-angle, rigid multibody system is derived, along with notation used, the reference configuration, the definitions of the joints between the bodies, generalized coordinates, and generalized speeds. This approach for dynamics is similar to that used by T. R. Kane (*Dynamics*, 3rd ed., 1978.)

5 Molecular Dynamics Simulation Architecture

The general system architecture 48 of the software and some of its processes for modeling molecules in accordance with the present invention are illustrated in Fig. 1. Each large rectangular block represents a software module and arrows represents information which passes between the software modules. The software system architecture has a modeler
10 module 50, a biochem components module 52, a physical model module 54, an analysis module 56 and a visualization module 58. The details of some of these modules are described below; other modules are available to the public.

The modeler module 50 provides an interface for the user to enter the physical parameters which define a particular molecular system. The interface may have a graphical
15 or data file input (or both). The biochem components module 52 translates the modeler input for a particular mathematical model of the molecular system and is divided into translation submodules 60, 62 and 64 for mathematical modeling the molecule(s), the force fields and the solvent respectively of the system being modeled. There are several modeler and biochem components modules available including, for example, Tinker (Jay Ponder,
20 TINKER User's Guide, Version 3.8, October 2000, Washington University, St. Louis, MO).

With the translated physical parameters from the biochem components module 52, the physical model module 54 defines the molecular system mathematically. At the core of the module 54 is a multibody system submodule 66. The physical model module 54 and multibody system submodule 66 are described below in detail. Co-pending application, U.S.
25 Appln. No. _____, entitled, "METHOD FOR ANALYTICAL JACOBIAN COMPUTATION IN MOLECULAR MODELING," and filed on even date, which claims priority from the previously referenced provisional patent applications and which co-pending application is incorporated by reference in its entirety, has further descriptions of the physical model module 54 and multibody submodule 66.

30 The analysis module 56, which communicates with the physical model module 54 and the visualization module 58, provides solutions to the computational models of the molecular systems defined by the physical model module 54. The analysis module 56 consists of a set of integrator submodules 68 which integrate the differential equations of the

physical model module 54. The integrator submodules 68 advance the molecular system through time and also provide for static analyses used in determining the minimum energy configuration of the molecular system. The analysis module 56 and its integrator submodules 68 contain most of the subject matter of the present invention and are described in detail below.

The visualization module 58 receives input information from the biochem components module 52 and the analysis module 56 to provide the user with a three-dimensional graphical representation of the molecular system and the solutions obtained for the molecular system. Many visualization modules are presently available, an example being VMD (A. Dalke, *et al.*, VMD User's Guide, Version 1.5, June 2000, Theoretical Biophysics Group, University of Illinois, Urbana, Illinois).

The described software code is run on conventional personal computers, such as PCs with Pentium III or Pentium IV microprocessors manufactured by Intel Corporation of Santa Clara, California. This contrasts with many current efforts in molecular modeling which use supercomputers to perform calculations. Of course, further speed improvements can be obtained by running the described software on faster computers.

MOLECULAR MODEL AND MULTIBODY SYSTEM DESCRIPTION

The integrators described below in the submodule 68 operate upon a set of equations which describe the motion of the molecular model in terms of a multibody system (MBS). To aid the computation of the integration methods described in detail below, a torsion angle, rigid body model is used to describe the subject molecule system, in accordance with the present invention. Internal coordinates (selected generalized coordinates and speeds) are used to describe the states of the molecule.

The MBS is an abstraction of the atoms and effectively rigid bonds that make up the molecular system being modeled and is selected to simplify the actual physical system, the molecule in its environment, without losing the features important to the problem being addressed by the simulation. With respect to the general system architecture illustrated in Fig. 1, the MBS does not include the electrostatic charge or other energetic interactions between atoms nor the model of the solvent in which the molecules are immersed. The force fields are modeled in the submodule 62 and the solvent in the submodule 64 in the biochem components module 52.

Fig. 2 illustrates the tree structure of the MBS of a subject molecule. The basic abstraction of the MBS is that of one or more collections of hinge-connected rigid

bodies 170. A rigid body is a mathematical abstraction of a physical body in which all the particles making up the body have fixed positions relative to each other. No flexing or other relative motion is allowed. A hinge connection is a mathematical abstraction that defines the allowable relative motion between two rigid bodies. Examples of these rigid bodies and hinge connections are described below.

One or more of the bodies, called base bodies 172, have special status in that their kinematics are referenced directly to a reference point on ground 174. The system graph is one or more "trees". An important property of a tree is that the path from any body to any other body is unique, i.e., the graph contains no loops. The bodies in the tree are n in number (the base has the label 1). The bodies in the tree are assigned a regular labeling, which means that the body labels never decrease on any path from the base body to any leaf body 176. A leaf body is one that is connected to only a single other body. A regular labeling can be achieved by assigning the label n to one of the leaf bodies 178 (there must be at least one). If this body is removed from the graph, the tree now has $n - 1$ bodies. The label $n - 1$ is then assigned to one of its leaf bodies 180, and the process is repeated until all the bodies have been labeled. This is also done for any remaining trees in the system.

To help maintain the relationship between the bodies, an integer function is used to record the inboard body for each body of the system. The inboard body for each base is ground and i , the parent or inboard body 182 for body k 184, is referred to as $i = \text{inb}(k)$. Additionally, the symbol N refers to the inertial, or ground frame 174. A superscript O refers to the ground origin (0,0,0).

The symbol for the vector from one point to another contains the name of the two points. Thus, r^{PQ} is the vector from the point P to point Q . A vector representing the velocity of a point in a reference frame contains the name of the point and the reference frame: ${}^N v^P$. Certain symbols to be introduced later relate two reference frames. In this case, the symbol contains the name of two frames. Thus, ${}^i C^k$ is the direction cosine matrix for the orientation of frame k in frame i . This symbol refers to the direction cosine matrix for a typical body in its parent frame. Thus, ${}^i C^k(j)$ indicates the actual body j in question. The left and right superscripts do not change with the body index. This is also true for the other symbols.

An asterisk indicates the transpose: $H^*(k)$, for example. A tilde over a vector indicates a 3 by 3 skew-symmetric cross product matrix: $\tilde{v}w \triangleq v \times w$. \underline{E}_i is an i by i identity matrix., and $\underline{0}_i$ is a zero vector of length i and $\underline{0}$ is an i by i zero matrix.

Rigid Bodies of the Model

Fig. 3 illustrates the reference configuration 190 of a sample “tree” of the MBS. More than one tree is allowed. A point of each body is designated as Q , its hinge point. For example point Q_k 186 is the hinge point for body k 184. A fixed set of coordinate axes is established in the inertial frame 198. An arbitrary configuration of the MBS is chosen as its reference configuration 190. While in this configuration the image of the inertial coordinate axes is used to establish a set of body-fixed axes in each body. In the reference configuration each hinge point Q is coincident with P , a point of its parent body (or extended body.) For each body, point P is called the body’s inboard hinge point. So the inboard hinge point P_k 188 for body k 184 is a point fixed in its parent body i 182. The inboard hinge point for each base body is a point O 192 fixed in ground. The expanded view that shown in Fig. 2 more clearly shows that point Q_k 186 is fixed in body k 184 and point P_k 188 is fixed in parent body i 182.

The hinge point locations define $\mathbf{d}(k)$ 194, a constant vector for each body, and can also be written r^{Q,P_k} . The vector for body k is fixed in its parent body i . It spans from the hinge point for body i to the inboard hinge point for body k . The vector $\mathbf{d}(1)$ 196 spans from the inertial origin to the first base body’s inboard hinge point (also a point fixed in ground), and can be written r^{O,Q_1} .

For a body, $m(k)$, $\mathbf{p}(k)$, and $\underline{\mathbf{I}}_{Q_k}(k)$ define the mass properties of body k for its hinge point Q_k . These are, respectively, the mass, the first mass moment, and the inertia matrix of the body for its hinge point in the coordinate frame of the body. For a rigid body made up of a distribution of particles, the mass properties are constants that are computed by a preprocessing module. The details of these computations can be found in standard references, such as Kane, T.R., *Dynamics*, 3rd Ed., January 1978, Stanford University, Stanford, CA.

Let $M(k)$, the spatial inertia of body k for its hinge point Q_k , be given by the symmetric 6 by 6 matrix

$$M(k) = \begin{bmatrix} \underline{\mathbf{I}}_{Q_k}(k) & \tilde{\mathbf{p}}(k) \\ -\tilde{\mathbf{p}}(k) & m(k)\underline{\mathbf{E}}_3 \end{bmatrix}$$

Each joint in the system is described by geometric data. For instance, a pin joint is characterized by an axis fixed in the two bodies connected by the joint. The particular data for a joint depends on its type. The number n , the *inb* function, the system mass properties, the vectors $\mathbf{d}(k)$, and the joint geometric data (including joint type) constitute the *system parameters*.

Joints and Generalized Coordinates of the Model

Fig. 4 illustrates the joint definitions of the preferred embodiment of the MBS: the slider joint 100, the pin joint 102, and the ball joint 104. Each joint allows translational or rotational displacement of the hinge point Q_k 106 relative to the inboard hinge point P_k 108. These displacements are parameterized by $q(k)$ 110, the generalized coordinates for body k . In passing, it should be noted that generalized coordinates are examples of generalized quantities, which refer to quantities that have both rotational character and translational character. For instance, a generalized force acting at a point consists of both a force vector and a torque vector. The generalized coordinate $q(k)$ for the slider joint 100 is the sliding displacement x 112. The generalized coordinate $q(k)$ for the pin joint 102 is the angular displacement θ 114. The generalized coordinate $q(k)$ for the ball joint 104 is the Euler parameters $(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)$ 116.

Each joint may be a pin, slider, or ball joint; or a combination of these joints. Many other joint types are possible, including, but not limited to, free joints, U-joints, cylindrical joints, and bearing joints. For instance, $q(k) = (x, y, z)$, the inertial measure numbers of the vector from the base body inboard hinge point to the base body hinge point express the base body displacement in ground as three orthogonal slider joints. A free joint consists of three orthogonal slider joints combined with a ball joint, and has the full 6 degrees of freedom.

The collection of generalized coordinates for all the bodies comprises the vector \mathbf{q} , the generalized coordinates for the system.

Given the generalized coordinates for a particular joint, two quantities: $r^{P_k Q_k}(k)$, the joint translation vector and ${}^i C^k(k)$, the direction cosine matrix for body k in its parent are formed. The translation vector $r^{P_k Q_k}(k)$ expresses the vector from the inboard hinge point P of body k to the hinge point Q of body k , in the coordinate frame of the parent body. Details of these computations depend on the joint type and can be easily derived. For

purposes of this description, access to a function that can generate $r^{P_k Q_k}(k)$ and ${}^i C^k(k)$ given the system generalized coordinates is assumed.

As introduced, the choice of hinge point for each body is arbitrary. However, judicious choice greatly simplifies matters. For instance, for pin joints the hinge point should be chosen as a point on the axis of the joint. For this choice points P and Q remain coincident for all values of the joint angle, so the joint translation is zero. If the point Q is chosen at a distance from the axis, points P and Q move relative to each other:

$$r^{P_k Q_k}(k) = \lambda \times r^{O_k} \sin \theta - (1 - \cos \theta) (\underline{\underline{E}}_3 - \lambda \lambda^*) r^{O_k}$$

where λ is the joint axis unit vector, θ is the joint angle, and r^{O_k} is the vector from any point on the axis to point Q.

For pin joints and ball joints, a point on the axis is always chosen as the hinge point. For these joints the translation vector $r^{P_k Q_k}(k)$ is zero.

For a slider joint, the translation vector $r^{P_k Q_k}(k)$ is $q(k)\lambda$.

The direction cosine matrix for a pin is

$${}^i C^k(k) = \underline{\underline{E}}_3 \cos \theta + \tilde{\lambda} \sin \theta + \lambda \lambda^* (1 - \cos \theta) .$$

The direction cosine matrix for a slider is $\underline{\underline{E}}_3$.

Generalized Speeds of the Model

Let ${}^i V^k(k)$, the generalized velocity of the hinge point of body k measured in its parent i , be parameterized by $u(k)$, a set of generalized speeds. Then:

$${}^i V^k(k) = \begin{pmatrix} {}^i \omega^k(k) \\ {}^i v^{Q_k}(k) \end{pmatrix} = H^*(k) u(k)$$

Here, the matrix $H(k)$ is called the joint map for this joint. It is a $n_u(k)$ by 6 matrix, where $n_u(k)$ is the number of degrees of freedom for the joint (1 for a pin or slider, 3 for a ball, 6 for a free joint). $H(k)$ can, in general have dependence on coordinates q . Given the generalized speeds for the joint, the joint map generates the joint linear and angular velocity, expressed in the child body frame. The following are used for the joints:

$$H(k) = \begin{bmatrix} \underline{\lambda} & 0 & 0 & 0 \end{bmatrix}, \text{ pin}$$

$$H(k) = \begin{bmatrix} 0 & 0 & 0 & \underline{\lambda} \end{bmatrix}, \text{ slider}$$

$$H(k) = \begin{bmatrix} \underline{E}_3 & \underline{0}_3 \end{bmatrix}, \text{ ball}$$

$$H(k) = \begin{bmatrix} \underline{E}_3 & \underline{0}_3 \\ \underline{0}_3 & {}^i C^k(k) \end{bmatrix}, \text{ free}$$

The collection of generalized speeds for all the bodies comprises the vector u , the generalized coordinates for the system. As before, access to a function that can generate the vector ${}^i V^k(k)$ given (q, u) and a specific joint type, is assumed. Access to a function that can compute the derivatives $\dot{q}(k) = \dot{q}(q(k), u(k))$ is also assumed. This routine generates the time derivative of the generalized position coordinates:

$$\dot{q} = W(q)u$$

where $W(q)$ is a block diagonal matrix that relates \dot{q} and u , with each block depending upon the joint type:

$\dot{q} = u$ for pin joint, slider joint

$$\begin{bmatrix} \dot{\varepsilon}_1 \\ \dot{\varepsilon}_2 \\ \dot{\varepsilon}_3 \\ \dot{\varepsilon}_4 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \varepsilon_4 & -\varepsilon_3 & \varepsilon_2 \\ \varepsilon_3 & \varepsilon_4 & -\varepsilon_1 \\ -\varepsilon_2 & \varepsilon_1 & \varepsilon_4 \\ -\varepsilon_1 & -\varepsilon_2 & \varepsilon_4 \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \text{ for ball joint}$$

where $q = [\varepsilon_1 \ \varepsilon_2 \ \varepsilon_3 \ \varepsilon_4]^*$ and $u = [\omega_1 \ \omega_2 \ \omega_3]^*$

and a free joint is a combination of 3 slider joints and one ball joint. Note that there are 4 \dot{q} 's (derivatives of the Euler parameters) associated with 3 u 's for ball joints.

Similarly, ${}^i A^k(k)$, the generalized acceleration of the hinge point of body k in its parent, is given by:

$${}^i A^k(k) = \begin{pmatrix} {}^i \alpha^k(k) \\ {}^i a^k(k) \end{pmatrix} = H^*(k) \dot{u}(k)$$

It is these generalized coordinates q , and generalized speeds u , the internal coordinates for purposes of this description, of the molecular system which are calculated. Rather than working with the typical inertial coordinates (x , y, z) and speeds in these inertial coordinate systems, calculations for the subject molecular system are reduced.

5 CALCULATIONS OF THE EQUATIONS OF MOTION

With the exemplary rigid multibody, torsion angle model described, the equations of motion can now be calculated. In accordance with the present invention, the motion of the MBS molecular model is determined by the Residual Form. The Residual Form method requires calculations termed the “first” kinematic calculations to distinguish
10 them from the “second” kinematic calculations, which are further required by the Direct Form (which is included in this description for purposes of comparison).

First Kinematic Calculations for the Molecular Model

In the first kinematic calculations, given the internal coordinates of the molecular system, (q, u, \dot{u}) and the system parameters, the following position, velocity and
15 acceleration kinematics are computed for each rigid body k of the molecular model. (In passing, it should be noted that when the First Kinematic calculations are done for the Residual Form method, the \dot{u} is passed in as a guess of the solution which the integration method then refines to the correct solution. In contrast, \dot{u} is set to zero when used for the Direct Form method. This is shown clearly in the later descriptions of the two methods.)

For each body k compute:

$$\begin{aligned} & {}^N C^k(k), r^{Q_i Q_k}(k), r^{O Q_k}(k), {}^i \phi^k(k), \\ & {}^N \omega^k(k), {}^N v^{Q_k}(k), V(k), \\ & {}^N \alpha^k(k), {}^N a^{Q_k}(k), A(k) \end{aligned}$$

These computations are done recursively, starting from each base body and progressing to the leaves.

${}^N C^k(k)$, the direction cosine matrix for body k in ground is defined as:

$$\begin{aligned} & {}^N C^k(1) = {}^i C^k(1) \\ & {}^N C^k(k) = {}^N C^k(i) {}^i C^k(k), \quad k = 2, \dots, n, \quad i = \text{inb}(k) \end{aligned}$$

${}^i C^k(k)$ comes from the joint routine described above.

$r^{Q_i Q_k}(k)$, the position vector from Q_i , the hinge point of the parent of body k to Q_k , the hinge point of body k , expressed in the parent frame, is defined as:

$$r^{Q_k}(k) = \mathbf{d}(k) + r^{P_k Q_k}(k), \quad k = 1, \dots, n$$

$r^{P_k Q_k}(k)$ comes from the joint routine.

$r^{O Q_k}(k)$, the position vector from the inertial origin O to Q_k , the hinge point of body k , expressed in the global frame, is defined

$$\begin{aligned} r^{O Q_k}(1) &= r^{O Q_k}(1) \\ r^{O Q_k}(k) &= r^{O Q_k}(i) + {}^N C^k(i) r^{Q_i Q_k}(k), \quad k = 2, \dots, n, \quad i = \text{inb}(k) \end{aligned}$$

${}^i \phi^k(k)$, the rigid body transformation operator for body k is defined

$${}^i \phi^k(k) = \begin{pmatrix} {}^i C^k(k) & \tilde{r}^{Q_i Q_k}(k) {}^i C^k(k) \\ \underline{0}_3 & {}^i C^k(k) \end{pmatrix}, \quad k = 1, \dots, n$$

$V(k)$, the spatial velocity for body k at its hinge point, expressed in the frame of body k , is defined

$$\begin{aligned} V(1) &\triangleq \begin{pmatrix} {}^N \omega^k(1) \\ {}^N v^{Q_k}(1) \end{pmatrix} = {}^i V^k(1) \\ V(k) &\triangleq \begin{pmatrix} {}^N \omega^k(k) \\ {}^N v^{Q_k}(k) \end{pmatrix} = {}^i \phi^{k*}(k) V(i) + {}^i V^k(k), \quad k = 2, \dots, n, \quad i = \text{inb}(k) \end{aligned}$$

$A(k)$, the spatial acceleration for body k at its hinge point, expressed in the frame of body k , is defined

$$\begin{aligned} A(1) &\triangleq \begin{pmatrix} {}^N \alpha^k(1) \\ {}^N a^{Q_k}(1) \end{pmatrix} = {}^i A^k(1) \\ A(k) &\triangleq \begin{pmatrix} {}^N \alpha^k(k) \\ {}^N a^{Q_k}(k) \end{pmatrix} = \bar{A} + \begin{pmatrix} \tilde{\omega} & \underline{0}_3 \\ \underline{0}_3 & 2\tilde{\omega} \end{pmatrix} {}^i V^k(k) + {}^i A^k(k), \quad k = 2, \dots, n, \quad i = \text{inb}(k) \end{aligned}$$

where

$$\begin{aligned} \bar{A} &= {}^i \phi^{k*}(k) A(i) + \begin{pmatrix} \underline{0}_3 \\ {}^i C^{k*}(k) ({}^N \omega^k(i) \times {}^N \omega^k(i) \times r^{Q_i Q_k}(k)) \end{pmatrix} \\ \omega &= {}^i C^{k*}(k) {}^N \omega^k(i) \end{aligned}$$

Of course, the computations can all be computed in a single pass if desired.

After completing these steps for one incremental time step, the MBS can service kinematics requests to compute the (generalized) position, velocity, or acceleration information for any point of any body. This is done by computing the required information for any point in terms of the hinge quantities for its body, using standard rigid body formulas.

Residual Computation

With the first kinematic calculations described above, the residual computation for the Residual Form method can be determined. This computation fills in two partitions of the vector $\begin{pmatrix} \rho_q \\ \rho_u \end{pmatrix}$ given previously. The first partition is called ρ_q , the kinematic residual, and the second partition is called ρ_u , the dynamic residual. The kinematic residual is computed from the difference between a \dot{q} , which is passed-in from the (implicit) integration submodules 66, and the derivative computed by each joint:

$$\dot{q} - W(q)u = \rho_q$$

The dynamics residual is also computed. Starting with a given state of the molecular model, i.e., given (q, u, \dot{u}) and the system parameters, a program routine models the 'environment' of the MBS. Such routines are readily available to, or can be created by, practitioners in the computer modeling field. The routine takes the values (q, u) determined by and passed in from the integration submodules 66 and returns (the state-dependent)

$T(k) = \begin{pmatrix} T_{Q_k}(k) \\ F(k) \end{pmatrix}$, the applied spatial force for a body k at its hinge point Q_k , and $\sigma(k)$, the

hinge torque for the body k . The dynamics residual, $\rho_u(k)$, associated with generalized speeds $u(k)$ for the body k is then computed by the following steps:

1. Perform the calculations for the molecular model by the Residual Form as described above with the passed-in state values (q, u, \dot{u}) ;
2. Generate $\hat{T}(k)$, the spatial load balance for each body k in the model having n bodies:

$$\hat{T}(k) = M(k)A(k) + \begin{pmatrix} {}^N\tilde{\omega}^k(k)({}^N\mathbf{I}_{Q_k}(k){}^N\omega^k(k)) \\ {}^N\tilde{\omega}^k(k)({}^N\omega^k(k) \times \mathbf{p}(k)) \end{pmatrix} - T(k)$$

$k = 1, \dots, n$

3. Compute $\rho_u(k)$

for $k = n$ to 2 by -1

$$\rho_u(k) = H(k)\hat{T}(k) - \sigma(k)$$

$$i = \text{inb}(k)$$

$$\hat{T}(i) += {}^i\phi^k(k)\hat{T}(k)$$

end

$$\rho_u(1) = H(1)\hat{T}(1)$$

The Residual Form method evaluates the extent to which the system differential equations are satisfied. Zero residual indicates that the applied forces are in balance with the inertia forces. However, this does not mean the system is in static equilibrium, but rather that the applied forces would reproduce the given \dot{u} when applied to the system in the state (q, u) . The residuals can be interpreted as that additional hinge torque needed to balance the applied and inertia forces. In the literature this method is known as either inverse dynamics, or the method of computed torques. It governs the case where the \dot{u} are all prescribed. At this point all the computations required for the Residual Form are complete. The residuals ρ_q and ρ_u are used directly by the implicit integrator in the integrator submodule 68.

To demonstrate the computational advantage of the Residual Form method, the Direct Form method for the equations of motion is also described with the same molecular model described above. The computational advantage is clearly shown by comparing operation counts of the Residual Form method and the Direct Form method for molecular models for molecules of different sizes.

Second Kinematics Calculations for the Molecular Model

To carry out the Direct Form method, calculations in addition to the first kinematics calculations are required. These additional calculations are termed the second kinematics calculations. The values $P(k), D(k), {}^i\psi^k(k), {}^iK^k(k)$ are computed as follows:

1. Perform the calculations for the Molecular Model by the Residual Form as described above, i.e., the first kinematics calculations.
2. $P(k)$, the articulated body inertia of each body k , is initialized.

$$P(k) = M(k), \quad k = 1, \dots, n$$

3. The objects below are then generated:

```

for  $k = n$  to 2 by -1
   $D(k) = H(k)P(k)H^*(k)$ 
   $G = P(k)H^*(k)D^{-1}(k)$ 
   $\bar{\tau} = \underline{\underline{E}}_6 - GH(k)$ 
   ${}^i\psi^k(k) = {}^i\phi^k(k)\bar{\tau}$ 
   ${}^iK^k(k) = {}^i\phi^k(k)G$ 
   $i = inb(k)$ 
   $P(i) += {}^i\psi^k(k)P(k){}^i\psi^{k*}(k)$ 
end
 $D(1) = H(1)P(1)H^*(1)$ 

```

The functional dependence of these quantities is only upon the generalized coordinate q . Therefore, the first kinematics calculations are programmed in anticipation of performing the second kinematics calculations.

5 Forward Dynamics Calculations

Finally, we can compute \dot{u} by sweeping inboard, then outboard:

```

 $z(k) = \underline{0}_6, \quad k = 1, \dots, n$ 
for  $k = n$  to 2 by -1
   $\varepsilon(k) = \rho_u(k) - H(k)z(k)$ 
   $\nu(k) = D^{-1}(k)\varepsilon(k)$ 
   $i = inb(k)$ 
   $z(i) += {}^i\psi^k(k)z(k) + {}^iK^k(k)\rho_u(k)$ 
end
 $\varepsilon(1) = \rho_u(1) - H(1)z(1)$ 
 $\nu(1) = D^{-1}(1)\varepsilon(1)$ 
 $\dot{u}(1) = \nu(1)$ 
 $\delta(1) = H^*(1)\nu(1)$ 
for  $k = 2$  to  $n$ 
   $i = inb(k)$ 
   $\delta(k) = {}^i\psi^{k*}(k)\delta(i) + H^*(k)\nu(k)$ 
   $\dot{u}(k) = \nu(k) - {}^iK^{k*}(k)\delta(i)$ 
end

```

With the First and Second Kinematics Calculations, and the Forward Dynamics Calculations, the Direct Form method is available.

Direct Form Method for the Equations of Motions

The Direct Form method takes the current state (q, u) and computes the derivatives (\dot{q}, \dot{u}) using the above algorithms, which are then used by the integration method to advance time.

- 5 Given: (q, u)
 Compute: (\dot{q}, \dot{u})
1. Compute \dot{q} using joint specific routine as above
 2. Perform above first kinematics calculations with $\dot{u} = 0$
 3. Generate residuals ρ_u as above
 - 10 4. Negate the residuals $\rho_u = -\rho_u$
 5. Perform second kinematics calculations
 6. Compute \dot{u} using forward dynamics step above

The Direct Form method produces the hinge accelerations \dot{u} in response to the applied forces acting on the system.

15 Fig. 5 summarizes the computation steps of the Residual Form method and the Direct Form method. It should be evident that since the Direct Form method includes the calculations of the Residual Form method, the Direct Form method is more computationally intensive than the Residual Form method. Stated differently, the Residual Form method requires less computer time to advance the molecular model in time.

20 Fig. 6 illustrates the computations required for the standard Direct Form method of the MD equations versus the Residual Form method. The operation count is for the preferred embodiment of using $\text{Order}(N)$ torsion angle formulation. Several size polypeptide molecules from 2 to 100 amino acid residues are shown. The operation count is reduced approximately by a factor of 7 for the Residual Form method.

25 Hence the present invention improves the speed with which accurate molecular dynamics simulations can be performed. The method allows a numerical integration algorithm to utilize a representation of the differential equations that requires fewer arithmetic operations to evaluate than previous methods. The Residual Form method of the equations includes $0 = M\dot{u} - f$ whereas the Direct Form method includes $\dot{u} = M^{-1}f$.

30 The Direct Form method requires evaluation of the state derivatives, and while this can be done efficiently using $\text{Order}(N)$ methods, the Residual Form can be computed with less cost. In addition, an analytical Jacobian of the residual equations can be formed at less cost than

the analytical Jacobian of the direct equations (see previously referenced co-pending application, U.S. Appln. No. _____, entitled, "METHOD FOR ANALYTICAL JACOBIAN COMPUTATION IN MOLECULAR MODELING," and filed on even date. A Jacobian is required by stable implicit integration methods and its formation is often the most time-consuming step in such methods.

The use of the Residual Form method can be applied to many forms of molecular modeling including, but not limited to:

- Constrained models of molecules with closed loops, as well as open tree structures. The constraint equations are typically adjoined to the equations of motion, such as, but not limited to: $M\dot{u} = f - A^T \lambda$, where A is the constraint matrix and λ are the Lagrange multipliers (note that A is not the same as the generalized acceleration ${}^iA^k(k)$, and λ is not the same as the vector λ which defines an MBS joint axis);
- Cartesian, or other formulations of the molecular models;
- Any order of the equations to be solved, including, but not limited to $\text{Order}(N)$, $\text{Order}(N^2)$, $\text{Order}(N^3)$, and $\text{Order}(N^4)$;
- All-atom models, rigid-body models, flexible-body models or combinations these models.

Additionally, the Residual Form method can work with many implicit integration methods, including, implicit Euler, Radau5, SDIRK3, SDIRK4, and other implicit Runge-Kutta methods, as well as DASSL and other implicit multistep methods.

With the Residual Form method according to the present invention, biomolecular systems, especially for protein and RNA including, but not necessarily limited to, folding, interactions with small drug-like molecules, and functions, such as conformational changes and binding, are effectively simulated.

Therefore, while the foregoing is a complete description of the embodiments of the invention, it should be evident that various modifications, alternatives and equivalents may be made and used. Accordingly, the above description should not be taken as limiting the scope of the invention which is defined by the metes and bounds of the appended claims.